

## Ingeniería en Computación

### Bases de Datos 1 – Planificador de Eventos

Profesora: Adriana Álvarez

### 1. Motivación

A la hora de organizar una reunión, fiesta, u otro tipo de actividad se debe pensar en todo tipo de factores como la comida, el lugar, los invitados, entre otros.

Hoy en día las personas ven la facilidad en organizar eventos por medio de WhatsApp, Facebook u otra red social, esto facilita la comunicación con los amigos y compañeros.

### 2. El problema

En la actualidad la mayoría de las personas que tienen un smartphone tienen redes sociales, sin embargo siempre hay una pequeña población que prefiere evitarlas, esto impide que puedan enterarse de ciertas actividades y reuniones de su interés.

Se cuenta con herramientas tales como los grupos de mensajería instantánea, pero esto puede llegar a ser muy tedioso e incómodo para los usuarios, ya que pertenecerían a cientos de grupos donde la información está dispersa y no son herramientas destinadas para organizar actividades adecuadamente.

### 3. Arquitectura

Estructure su aplicación como un Sistema N Capas. Investigue más detalles sobre esta arquitectura, de modo que

cuenta con al menos los componentes básicos ilustrados en la Figura 1. A continuación se da una descripción general de cada una de las capas:

- *User Interface (UI)*: capa de presentación, encargada de mostrar las vistas de interfaz gráfica mediante la cual los usuarios interactúan con la aplicación. Esta capa no toma decisiones ni procesa lógica de negocio, solamente se encarga de mostrar o capturar información.
- *Business Layer (BL)*: capa de lógica de negocio, encargada de modelar las reglas de negocio de la aplicación, toma las decisiones, cálculos, procesos, etc.
- *Data Access (DA)*: capa de acceso a datos, abstrae la comunicación del sistema con la base de datos, no debe confundirse con la base de datos como tal.

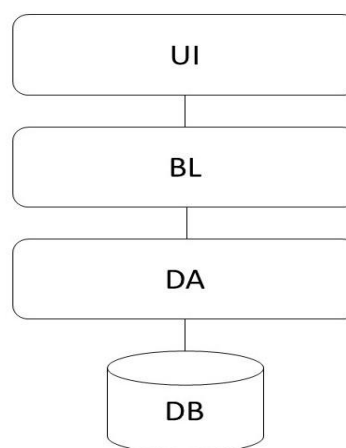


Figura 1 – Arquitectura N-Capas

Es probable que durante el desarrollo del sistema requiera de la

implementación de capas adicionales. Durante la revisión se deberá justificar las decisiones hechas sobre el diseño de la aplicación. Adicional a esto se espera una implementación elegante, con un código fuente que siga las normas básicas establecidas en el libro Clean Code (Martin, 2008).

## 4. Requerimientos

### a. Registro

El sistema debe permitir lo siguiente:

- a) Administradores para las operaciones de seguridad y administración de cada uno de los eventos.
- b) Tipo de usuario para las personas que deseen utilizar la aplicación., por ejemplo: profesores, estudiantes, funcionarios, público en general, entre otros.
- c) Registro de personas que ingresan al sistema con al menos nombre, apellidos, fecha de nacimiento, edad, foto, correo electrónico y teléfono.
- d) Cada persona que se conecta al sistema debe hacerlo por medio de un usuario y clave. La clave debe almacenarse encriptada.
- e) Deben existir usuarios Administradores para registrar la información de catálogos y manejar todo lo que pertenece a los eventos.
- f) Todos los datos pueden ser editados considerando los permisos de seguridad.
- g) Los administradores de eventos podrán definir si este es privado o público.
- h) Los usuarios podrán definir su estado en un evento como "Me interesa", "Asistiré", "Tal vez

asista" o "No asistiré" en cada uno de los eventos a los que han sido invitados.

- i) En cada evento existirá un muro en el cual los invitados podrán postear con comentarios o fotos cualquier cosa que consideren necesaria para el evento.
- j) Cuando el evento ya haya pasado los usuarios que hayan asistido podrán dejar una reseña en el mismo.

### b. Registro de Eventos

El sistema debe permitir lo siguiente:

- a) Clasificar los tipos de eventos. Por ejemplo educativas, recreativas, reunión, partido, entre otros.
- b) Registro de nombre del evento, tipo o clasificación, privacidad (Si es público o privado), descripción, fecha y hora, lugar, foto, y cualquier otro atributo que considere conveniente.
- c) Invitados: el sistema debe permitir invitar a otros usuarios, por medio del correo electrónico, en caso de que el evento sea **público** cualquier persona puede buscarlo y asistir a este.
- d) Cuando el evento ya haya pasado, **solamente los usuarios que asistieron** podrán dejar una reseña con estrellas y un comentario de qué les pareció la actividad.

### c. Publicaciones en los eventos previos a realizarse.

- a) Debe aparecer el nombre de la persona que realiza el post.
- b) Texto o foto del comentario.

#### d. Favoritos

El sistema debe permitir a los usuarios optar por las categorías favoritas, de forma tal que se le muestren con prioridad en la página de inicio los eventos públicos que pertenezcan a las categorías seleccionadas.

Este punto no limita que pueda visualizar el resto de la información.

#### e. Auditoría y bitácora

El sistema debe administrar campos de auditoría para los registros de datos en las tablas de la base de datos y para la última actualización de un registro.

Además, debe tener una bitácora que registre cada vez que un evento finaliza.

Utilice *triggers* para cumplir con esta funcionalidad donde se requiera.

Adicionalmente, en este módulo debe incluir una consulta para administradores de los eventos pasados, y se debe de poder filtrar por fecha.

#### f. Control y Usabilidad

El sistema permite a los administradores las operaciones de crear eliminar y modificar eventos, pueden invitar a otros usuarios, editar, eliminar o inactivar registros principalmente de los catálogos.

El sistema permite a los usuarios las operaciones de agregar un evento público a su lista de interés, aceptar invitaciones a eventos, comentar en ellos cualquier detalle de interés, y dejar una reseña cuando este evento haya finalizado.

Todos los catálogos deben registrarse, editarse y consultarse por medio del sistema. Se considera **no aceptable** el registro de datos por medio de la base de datos.

#### g. Módulo de Consultas

Este módulo les permitirá a los usuarios y administradores visualizar información precisa e inmediata.

Todas las consultas deben contener filtros de los datos que se solicitan.

Todas las consultas deben permitir combinación de datos en los filtros y filtros en blanco.

Las consultas mínimas que debe ofrecer el sistema son:

Consultas para administradores de eventos:

- a) Ver un listado de usuarios que se encuentran invitados a sus eventos, filtros mínimos estado de los usuarios, por ejemplo: "Me interesa", "Asistiré", "Tal vez asista" o "No asistiré".
- b) Listado de eventos realizados, y eventos próximos, filtros mínimos: fecha.
- c) Listado de reseñas por evento.

Para usuarios:

- a) Listado de eventos a los que se encuentra invitado, filtro: categoría del evento.
- b) Eventos públicos disponibles a los cuales podría asistir.
- c) Listado de los eventos a los que asistió y su respectiva reseña(si aplica).

#### h. Módulo de Estadísticas

Este módulo les permitirá a los administradores tener información precisa e inmediata de su información.

Cada una de las estadísticas deben mostrarse con datos numéricos, porcentajes y gráficos. Por ejemplo:

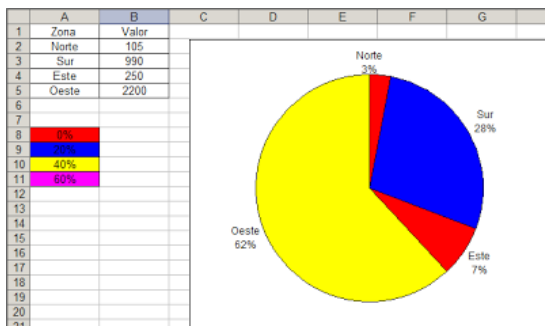
Todas las estadísticas deben permitir combinación de datos en los filtros.

Las estadísticas mínimas que debe ofrecer el sistema son:

- Total y porcentaje de eventos por categoría.
- Total y porcentaje de personas por categoría.
- Total y porcentaje de eventos por fecha. Filtros mínimos: día, semana, mes, año.
- Top n de eventos con mayor asistencia.
- Top n de eventos mejor calificados.
- Top n de eventos peor calificados.
- Rango de edad y porcentaje de usuarios. De 0 a 18, de 19 a 30, de 30 a 45, de 46 a 55, de 55 a 65, de 66 a 75, de 76 a 85, mayores a 85.
- Total y porcentaje de reseñas por categoría.

#### i. Jobs

El sistema debe generar un reporte al día con un recordatorio para los usuarios de los próximos eventos a los que asistirá, estos deben aparecer en orden cronológico.



Este reporte debe generarse como un Job de base de datos que corre todos los días a la 1 am de lunes a domingo.

## 5 Otras consideraciones

- El sistema debe presentarse en la defensa con datos **reales y bien clasificados**.
- Los catálogos deben tener los datos previamente registrados para la defensa del proyecto. Al menos 15 eventos, 10 personas que asistan a alguno de estos, toda la información de catálogos.
- Todos los campos que se ingresen en el sistema deben tener su respectiva salida (consulta, estadística, reporte, otro que considere).
- Debe utilizar **mySQL** o MariaDB para desarrollar el proyecto.
- Usted debe identificar donde aplica crear tablas catálogo para que el sistema sea flexible. Se calificará diseño.
- El sistema debe cumplir con los atributos de calidad de usabilidad, flexibilidad, fácil modificación, escalabilidad, entre otros.
- La aplicación debe ser gráfica.
- La aplicación debe ser de escritorio. **(Opcional: hacer la aplicación Web, no se darán puntos extra por hacerlo de esta manera).**
- Se evaluará el diseño y la correcta identificación de los parámetros.
- La aplicación debe tener un nombre y logo creado por el equipo. No puede ser un logo copiado de internet o de otras personas y debe ser original.

Puede solicitar ayuda de un estudiante de diseño.

11. El código debe trabajarse desde un programa de control de versiones y se deberá demostrar en la defensa la cantidad de commits realizados por cada miembro del equipo. Se recomienda por ejemplo Gitlab, Github.
12. El equipo puede estar conformado por grupos de un número máximo según le indique el profesor.
13. La entrega oficial se debe hacer en un zip en el tecDigital antes de la fecha y hora ahí indicada. No se aceptarán trabajos posteriores a esa fecha y hora. En caso de problemas de conexión, se debe aplicar el plan B de la presentación de Reglas del Juego.
14. La entrega debe contener los archivos fuente y el ejecutable de la aplicación (si aplica).
15. La documentación debe entregarse en pdf, latex, doc. No se acepta un link al documento.
16. Cualquier sospecha de copia anulará el trabajo y se procederá con el trámite administrativo.
17. El mejor proyecto tendrá +1 punto extra.

## 6 Documentación

La siguiente documentación debe ser entregada:

1. Diagrama de entidad – relación (modelo conceptual).
2. Diagrama Lógico.
3. Manual de usuario con la descripción del propósito y uso del sistema que incluya imágenes significativas de la aplicación. Si utiliza un wiki para la documentación, esta debe accederse desde la aplicación y el wiki completo debe ser entregado en el tecDigital.
4. Diccionario de datos.
5. Documento de matriz de casos de prueba (en Excel) con al menos 50 casos de prueba con los cuales se probará el programa. Debe contener un id como enumerador del caso de prueba, la descripción del caso de prueba, el resultado esperado y el estado (Certificado, Con Error, Pendiente). Además de una tabla pivote con la cantidad de casos de prueba en cada estado y el total.
6. El código debe estar documentado internamente según Clean Code. Los procedimientos, funciones o paquetes importantes deben ir documentados con descripción, el autor principal, fecha de creación.